1.**Full stack development**

**Full stack development:** It refers to the development of both **front end**(client side) and **back end**(server side) portions of web application.

**Full stack web Developers:** Full stack web developers have the ability to design complete web applications and websites. They work on the frontend, backend, database and debugging of web applications or websites.

**1.1 steps to connect the .html file to the database**

**STEP 1:**

In MYSQL workbench

Create a table on MYSQL workbench based on the front-end requirements.

**STEP 2:**

In visual studio code

Create “node.js” files in **music-js-sql** or **login-register-API** folder and check whether the code is running properly or not, if errors take place resolve the issue.

**STEP 3:**

In postman

Check whether data is inserting into the database or not.

**STEP 4:**

Open the corresponding, database html file i.e,music.html /login.html/ register.html

By using “ajax” and API URL, in order to connect to the database.

**STEP 5:**

Finally open the music.html/login.html/register.html page in web-browser

Insert the data in feedback form and click on submit button

On opening , MYSQL workbench,we can see that data is successfully inserted into the database table.

1.2 scripting

JavaScript

On the JavaScript side you should learn at least one modern framework:

● React.js

● Angular.js

● node.js

Maybe the popularity of jQuery has passed the top, but it is still the most used JavaScript framework.In present project we are using **node.js** framework.

**1.3 Database**

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

**2. System Requirements:**

**2.1 Hardware Requirements**

Processor : Any Update Processor

Ram : Min 2 GB

Hard Disk : Min 100 GB

**2.2 Software Requirements**

Operating System : Windows family

Technology : Java (1.7/1.8), node.js

Web Technologies : Html, Html-5, JavaScript, CSS

Web Server : MySQL,postman

Database : My SQL8.0 workbench

**3. PROJECT NAME**

**“Online Music System”**

**3.1 Project Name: Give information about the project**

The main aim of creating this Online Music System is html,css based web application it’s a collection of patriotic audio songs of different languages in one place where users can get based on the year also play and listen to the songs in our website at free of cost only is to provide a user friendly tool for music websites. This is one type of online music player. Most of the websites now a day’s Music products online but download music files for free of cost.

**3.2 Technologies Used:**

Visual studio code,node js, postman, My SQL workbench

**3.3 Technical Details:**

# **Software Configuration**

Operating System : Windows Family

Application Server : MySQL Server

Front End : HTML, CSS,BOOTSTRAP

Scripts : JavaScript.

Server side Script : Java Server Pages.

Database : Mysql 8.0

Database Connectivity : MYSQL workbench

**4. MODULES:**

**Admin**

Admin will collect all the Data for Our Online Music System like Songs ,feedback.

**Existing User**

Existing User will login in to our website by giving his username & password in to our Online Music System & access all the patriotic songs as per his/her requirement based on his wish and he can provide feedback.

**New User**

If the new person who is coming to our Online Music System has to register in our website by giving his/her basic information like ner username and new password there will be confidence in our database to keep a user name & password for next login .

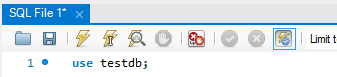
**5. Workflow**

**5.1 workflow of the feedback form in music.html**

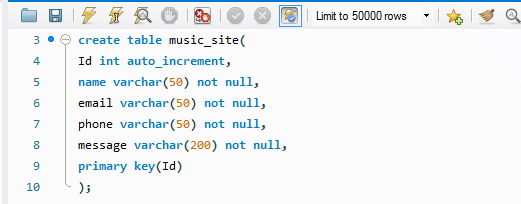
These screenshots will clearly explain how to connect feedback data to the MYSQL workbench.

**Backend of music.html page.**

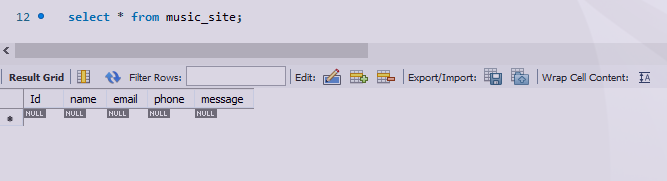
**!!STEP 1!!**



**Fig 5.1.1**



**Fig 5.1.2**



**Fig 5.1.3**

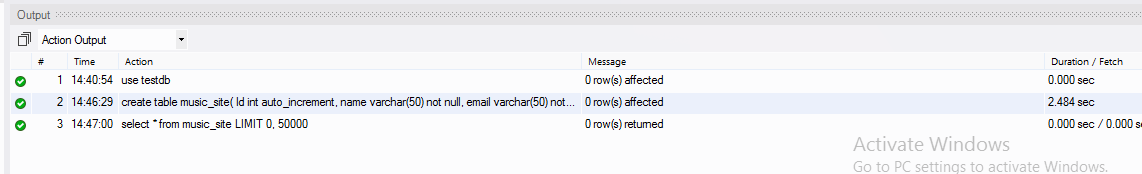


Fig 5.1.4

Now we have to create node.js files after installing node js in visual studio code:

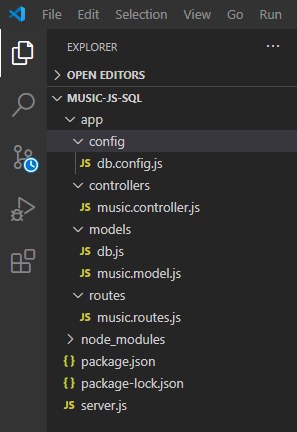
**npm install**  command is used to install node js on visual studio code

**!!STEP 2!!**

Files in node.js:

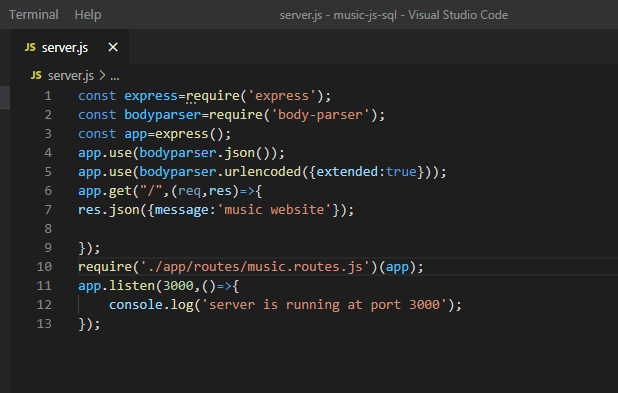
Code of node.js files.

In **music.js.sql folder**:



**Fig 5.1.5**

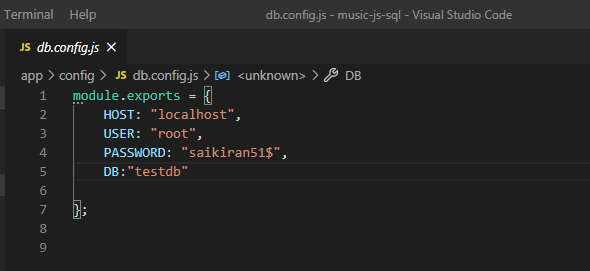
**server.js**

 **Fig 5.1.6**

In app folder:

config->

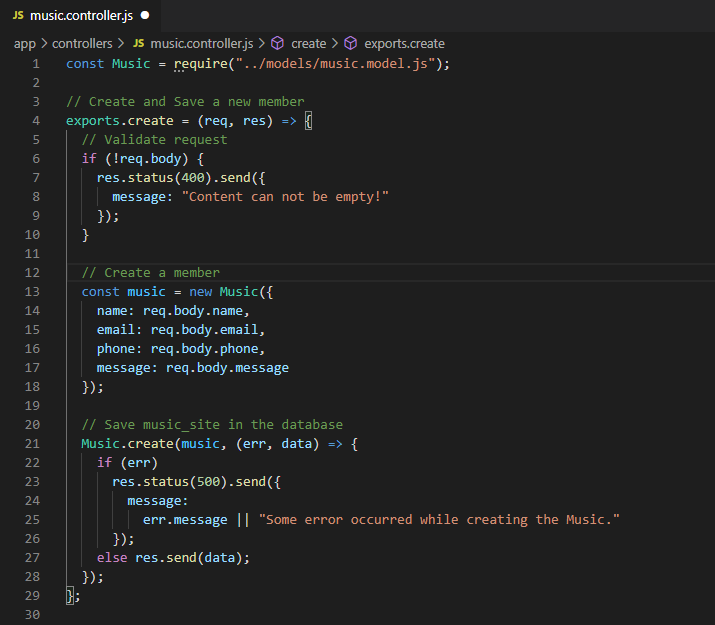
**db.config.js**



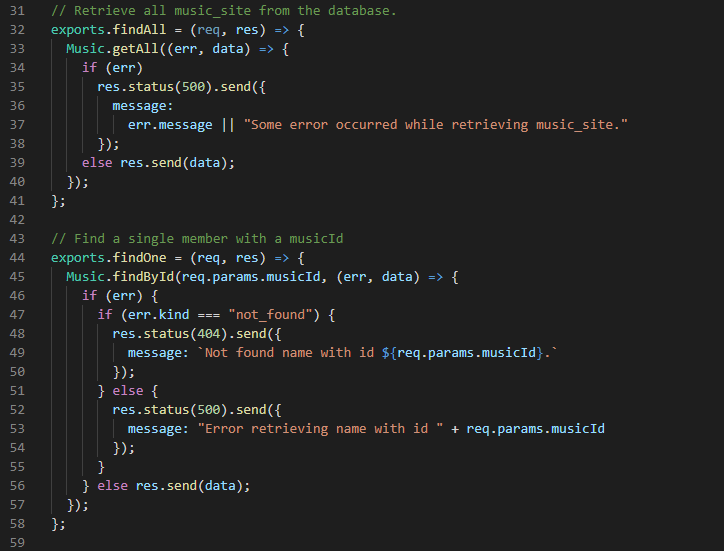
**Fig 5.1.7**

controllers->

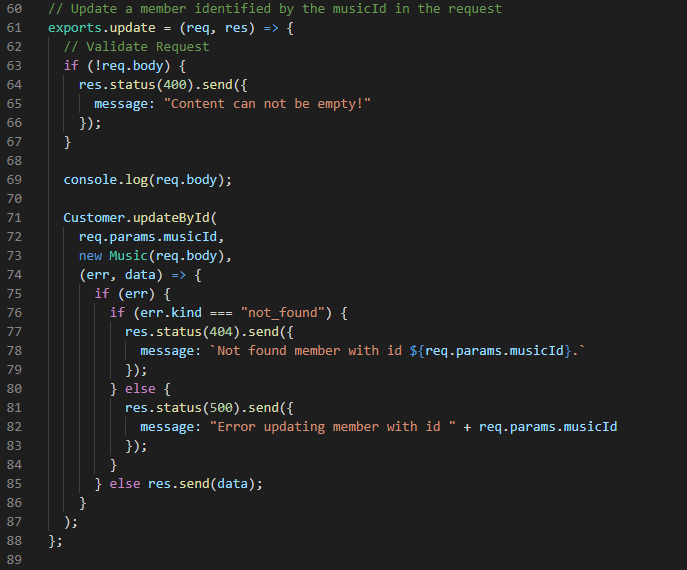
**music.controller.js**



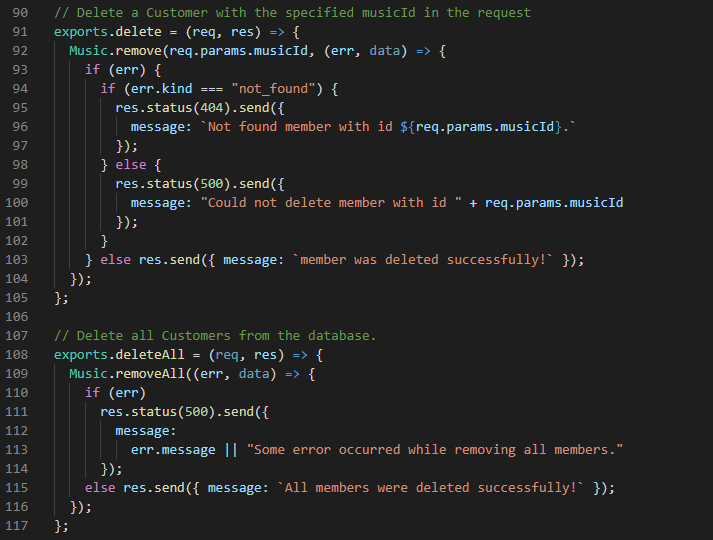
**Fig 5.1.8**



**Fig 5.1.9**



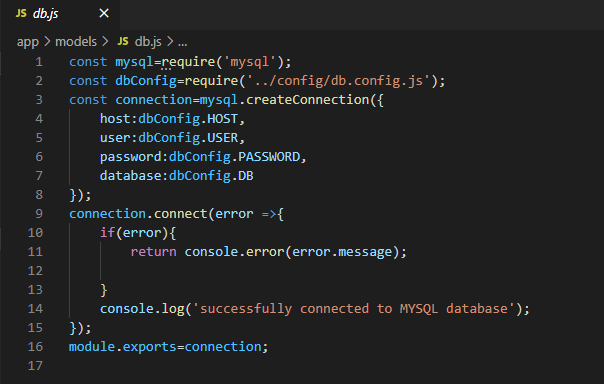
**Fig 5.1.10**



**Fig 5.1.11**

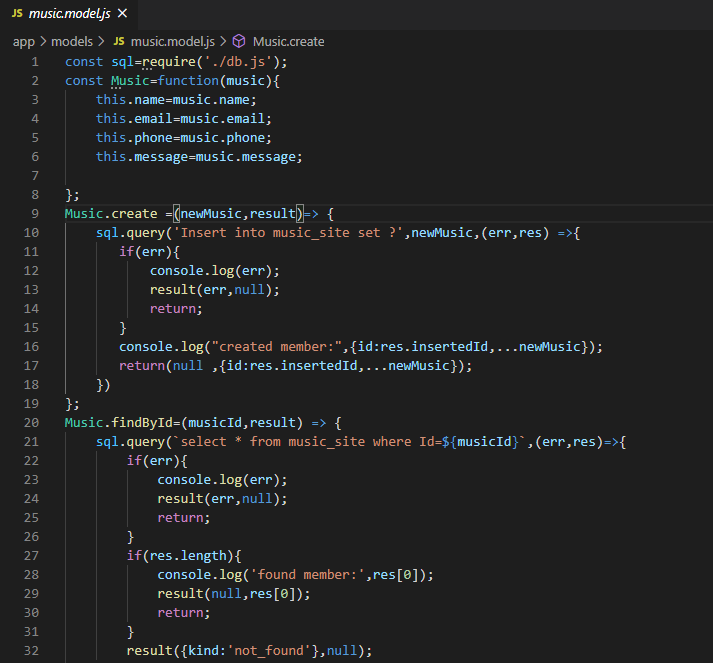
models->

**db.js**

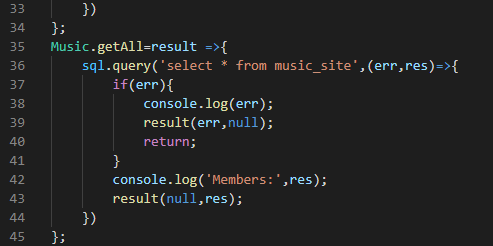


**Fig 5.1.12**

**music.model.js**

****

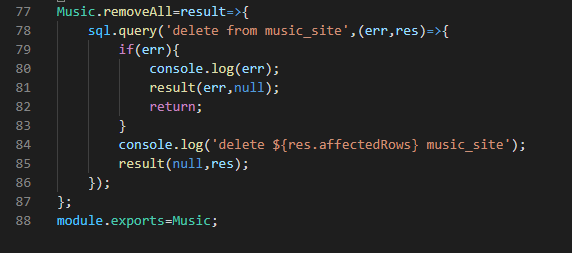
**Fig 5.1.13**

****

**Fig 5.1.14**

****

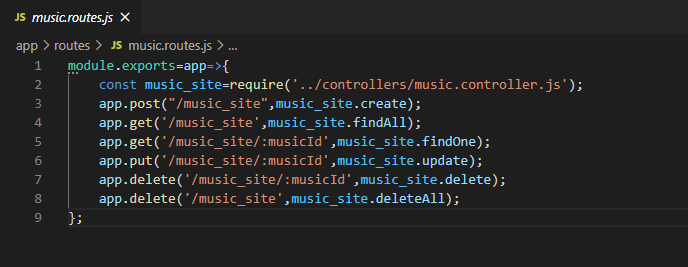
**Fig 5.1.15**



**Fig 5.1.16**

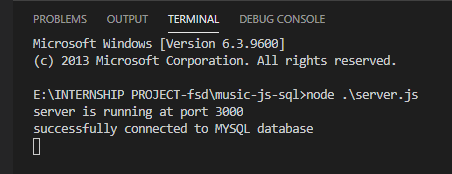
routes->

**Music.routes.js**

****

**Fig 5.1.17**

**Open new terminal and check weather the code is running properly or not:**

****

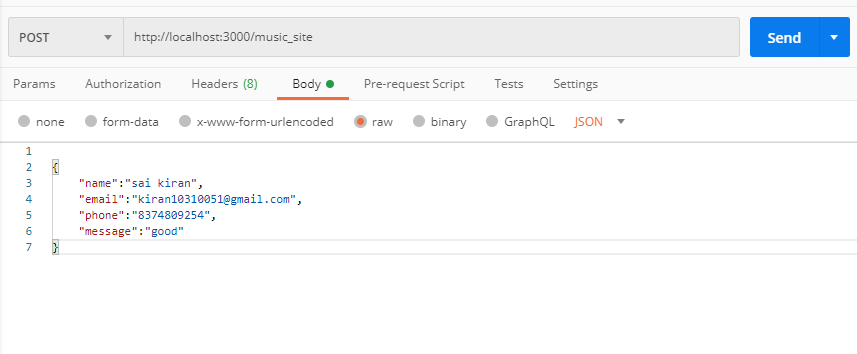
**Fig 5.1.18**

**!!STEP 3!!**

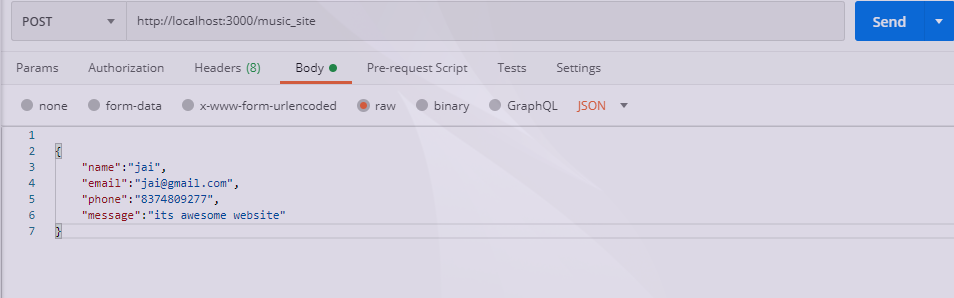
We use postman for testing purposes, whether POST requests are running properly or not.

**In postman:**

after creating post request:

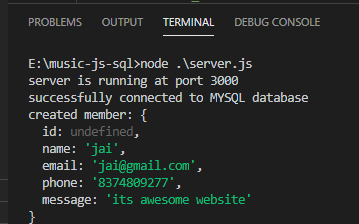


**Fig 5.1.19**



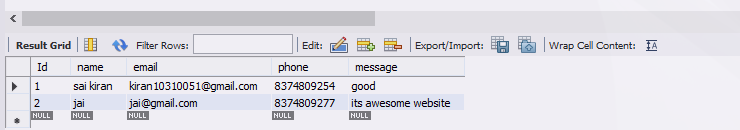
**Fig 5.1.20**

In visual studio code:



**Fig 5.1.22**

Sql database after updation:



**Fig 5.1.23**

**!!STEP 4!!**



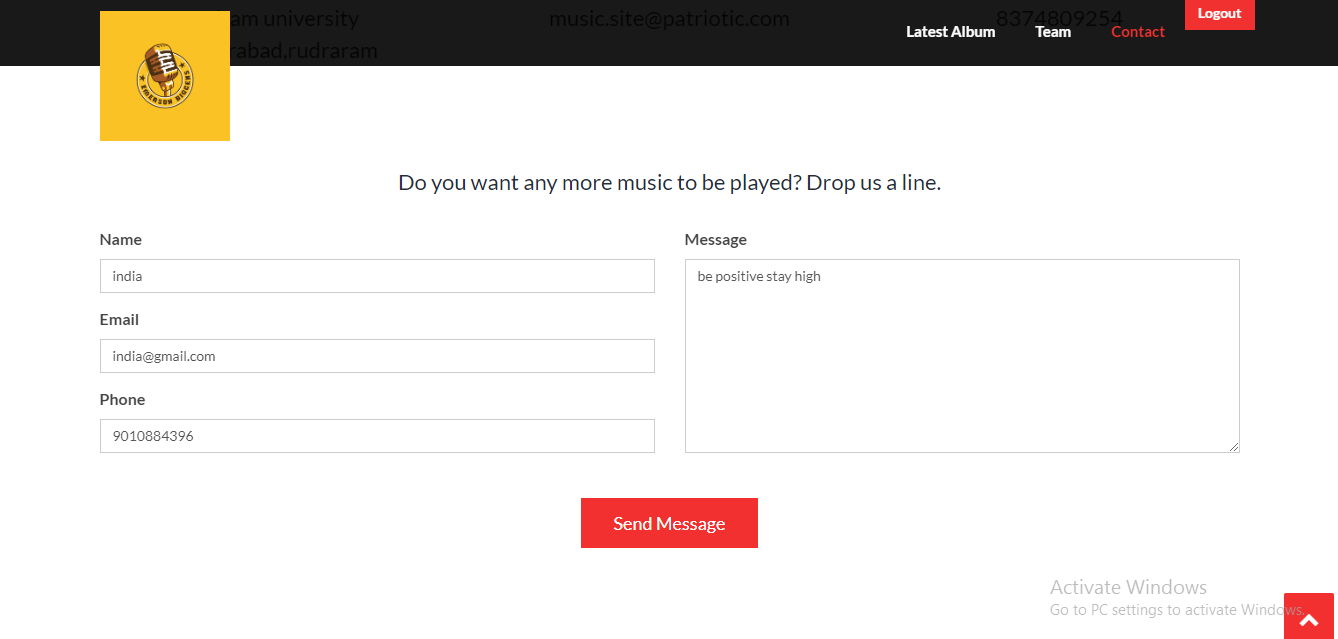
**Fig 5.1.24**

**!!STEP 5!!**

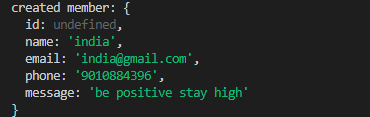
Now inserting values to “testdb” database, to SQL

Open **music.htm**l file

Now sending message to the server side:



**Fig 5.1.25**



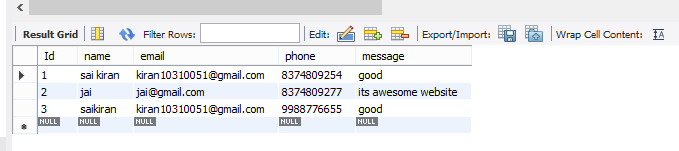
**Fig 5.1.26**

We can see a message that a member is successfully created in “visual studio code”

Now checking weather that values are inserted in MYSQL database or not:

Open MYSQL workbench:

Database present before:

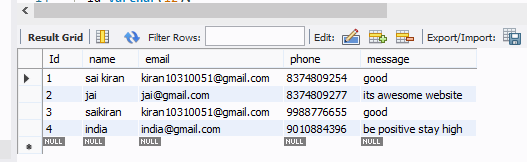


**Fig 5.1.27**

Now checking values are inserted or not:



now , the updated database is:



**Fig 5.1.28**

Now, the values are successfully inserted into the database from the front- end application.

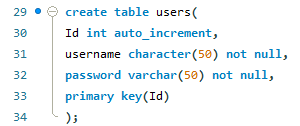
**5.2 workflow of registering a new user and logging a new user by checking details from the database.**

**Backend of login.html and register.html page:**

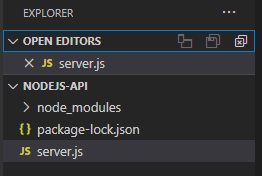
**Backend of registration page:**

Create table in MYSQL:





nodejs-API file on registration page:

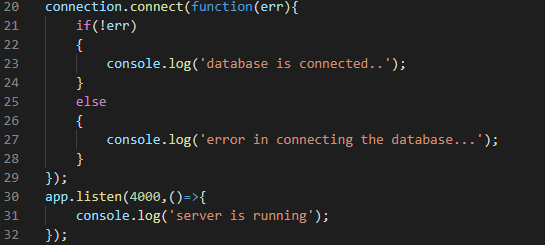


**Fig 5.2.1**

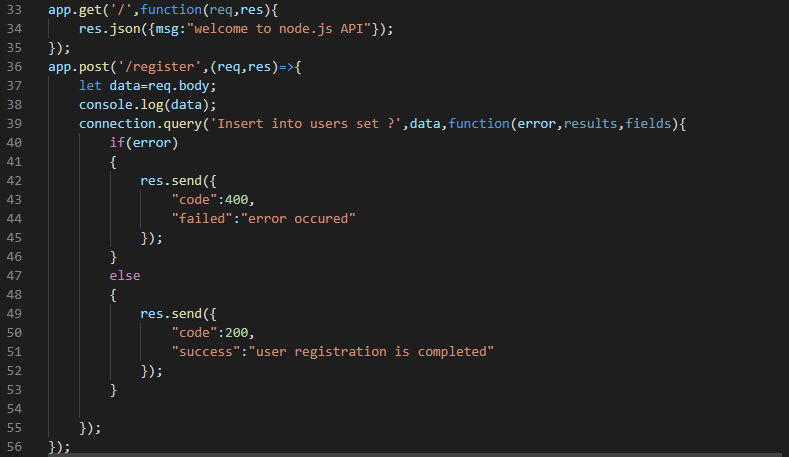
In server.js file:



**Fig 5.2.2**

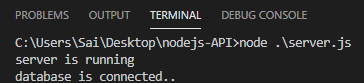


**Fig 5.2.3**



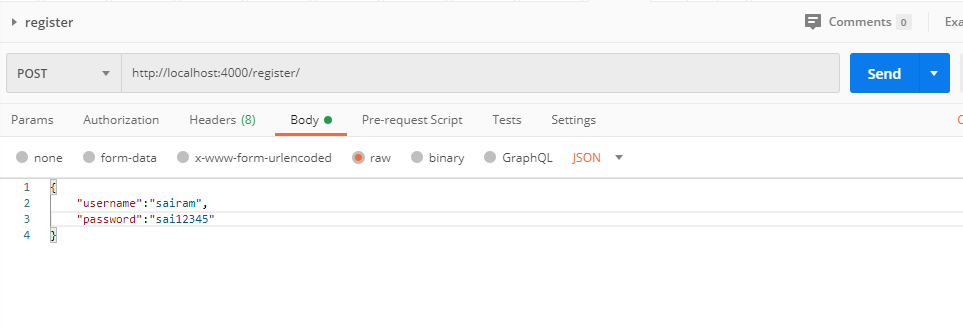
**Fig 5.2.4**

Now check whether code is running properly or not:



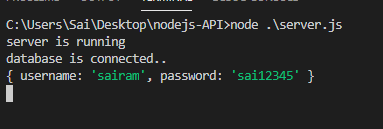
**Fig 5.2.5**

Now checking with postman:

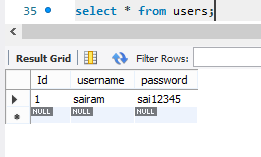


**Fig 5.2.6**

Now check in visual studio code and mysql weather data is inserted or not



**Fig 5.2.7**



**Fig 5.2.8**

Data is inserted successfully:

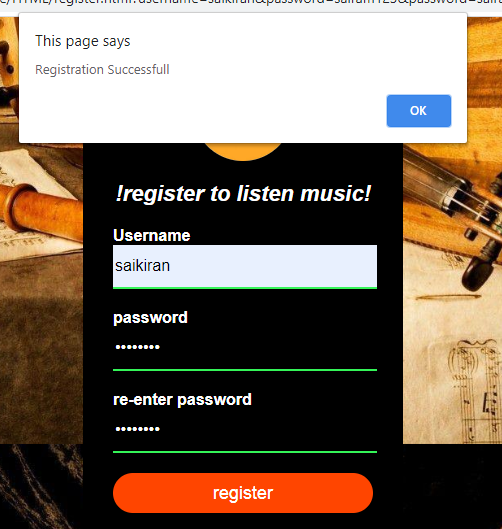
Now inserting data from **register.html** page:

Give api connection to register.html page:



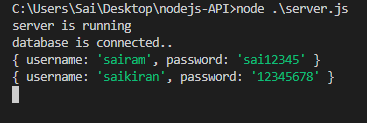
**Fig 5.2.9**

Now, inserting the data from login.html page:

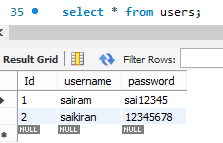


**Fig 5.2.10**

Now checking the database part:



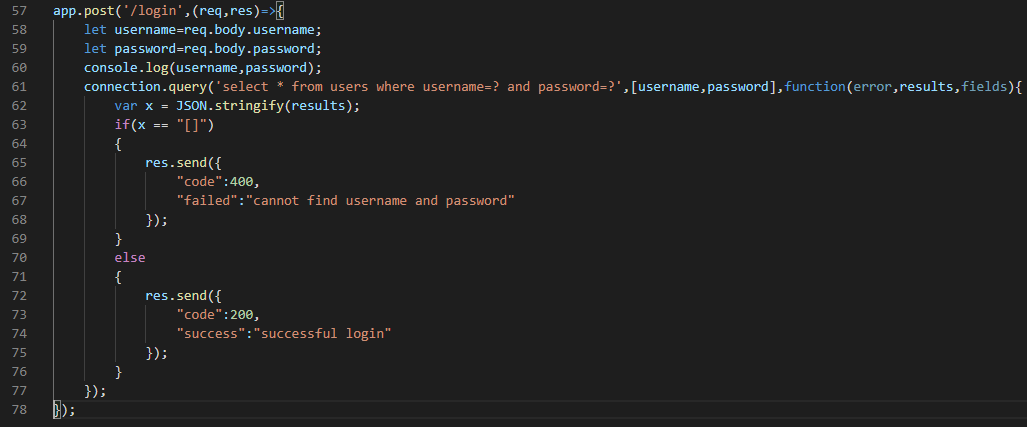
**Fig 5.2.11**

**Fig 5.2.12**

Data is successfully inserted into the database

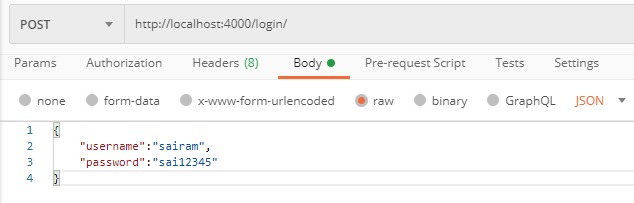
**Back-end of login.html page**

In server.js file:

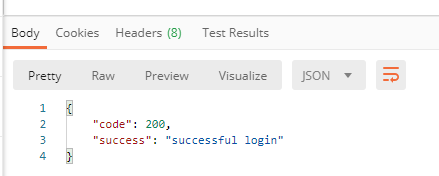


**Fig 5.2.13**

Now checking in postman whether data is validated or not:

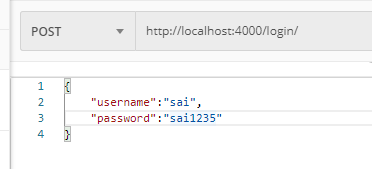


**Fig 5.2.14**



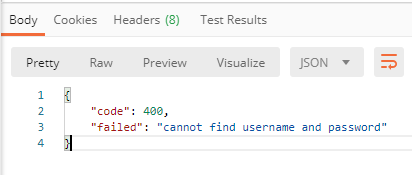
**Fig 5.2.15**

Now inserting some invalid data



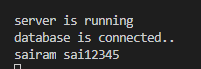
**Fig 5.2.16**

We can see the the entered data is invalid



**Fig 5.2.17**

Now checking the database part:



**Fig 5.2.18**

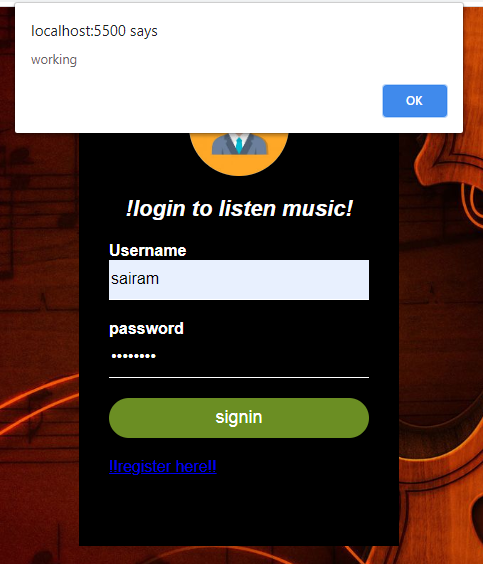
Now connecting API with login.html page:



**Fig 5.2.19**

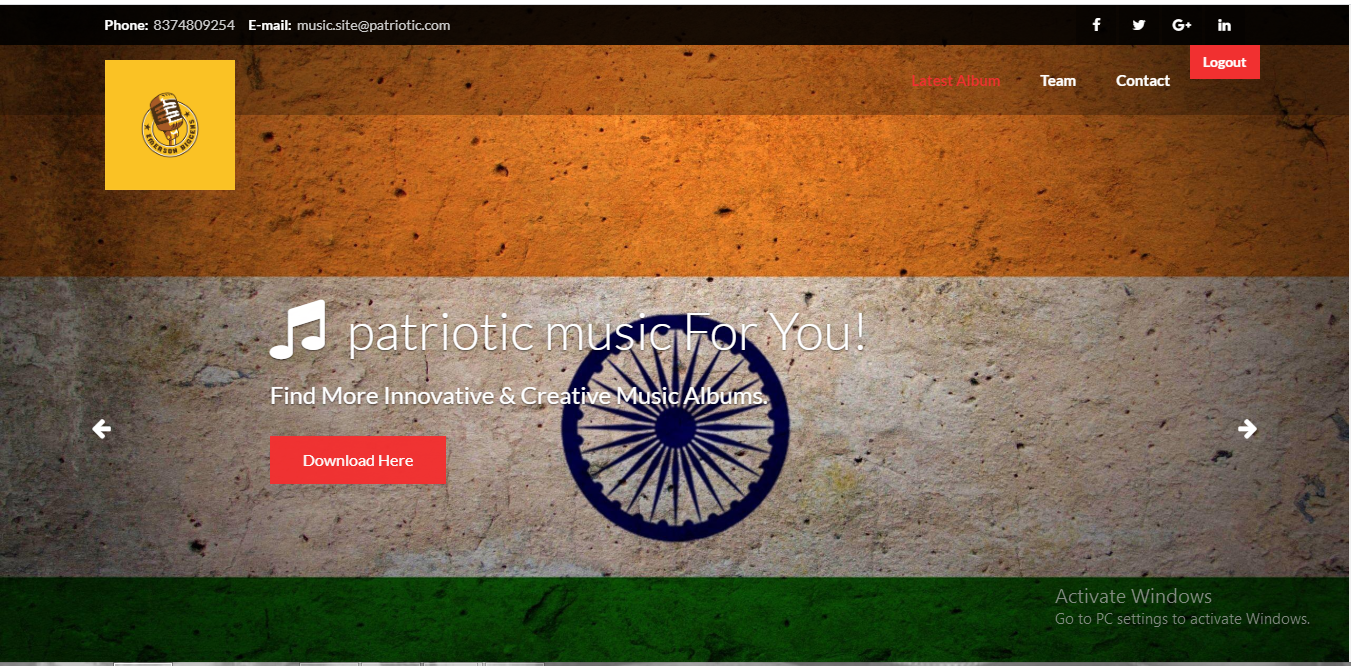
Now entering the dat in login.html page:

Now we have run**” LIVE SERVER PREVIEW”** in visual studio code.

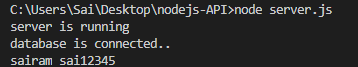


**Fig 5.2.20**

And it will redirect to the **music.html** page

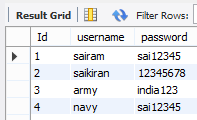


**Fig 5.2.21**

**Fig 5.2.22**

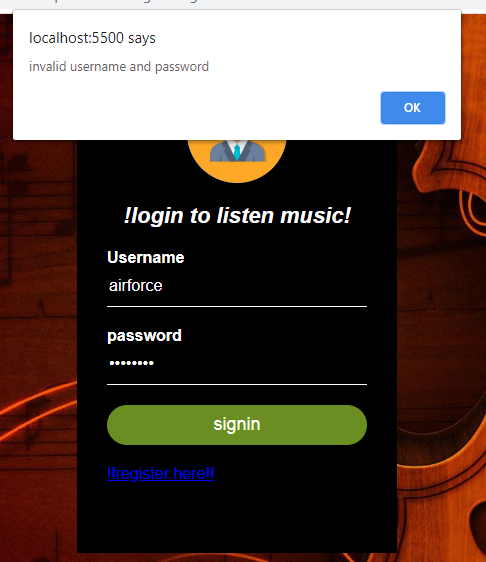
Now giving invalid details which is not present in the MYSQL database

Table shows the data which is already present.



**Fig 5.2.23**

Now entering invalid details which are not present in the database:



**Fig 5.2.24**

**6. MAINTENANCE**

Website maintenance details that one must keep a check on

● Scan for vulnerabilities. Security should be the primary reason for the maintenance of the website.

● Repairs and fixes.

● Browser compatibility testing.

● Software update.

● Website backups.

● Site speed

● Search engine optimization.

● Analytics.

**7. FUTURE SCOPE AND FUTURE ENHANCEMENT**

In Future we can play the online music Player. Most of the websites nowadays use Music products online.

**8. CONCLUSIONS**

Our Online Music System Java predicated web application it’s an accumulation of audio musical compositions of different languages in one place where users can get predicated on the year with play and Listen the musical compositions on our website at free of cost only is to provide cordial implementation for music websites.