**FULL STACK DEVELOPMENT**

(Online Music System)

*Summer Internship Report Submitted in partial fulfillment*

*of the requirement for undergraduate degree of*

**Bachelor of Technology**

In

**Computer Science and Engineering**

By

**Pulaparthi Sai Kiran**

**221710310051**

[**https://github.com/221710310051/FULL-STACK**](https://github.com/221710310051/FULL-STACK)

*Under the Guidance of*

Assistant Professor



Department Of Mention your Branch Name

GITAM School of Technology

GITAM (Deemed to be University)

Hyderabad-502329

July 2020

i

**DECLARATION**

I submitted this industrial training work entitled **“Online Music System** ” to GITAM (Deemed To Be University), Hyderabad in partial fulfillment of the requirements for the award of the degree of “**Bachelor of Technology**” in “**Computer Science and Engineering** ”. I declare that it was carried out independently by me under the guidance of **(specify the name of the professor)**, Asst. Professor, GITAM (Deemed To Be University), Hyderabad, India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: HYDERABAD Pulaparthi Sai Kiran

Date: 221710310051

ii

GITAM (DEEMED TO BE UNIVERSITY)

Hyderabad-502329, India

Dated:

**CERTIFICATE**

This is to certify that the Industrial Training Report entitled **“Online Music System ”** is being submitted by Pulaparthi Sai Kiran (221710310051) in partial fulfillment of the requirement for the award of **Bachelor of Technology** **in Computer Science and Engineering** at GITAM (Deemed To Be University), Hyderabad during the academic year 2019-20

It is faithful record work carried out by her at the Computer Science and Engineering, GITAM University Hyderabad Campus under my guidance and supervision.

**Mr. M. Venkateswarlu Dr.S.Phani Kumar**

Assistant Professor Professor and HOD

Department of CSE Department of CSE

**Please this space for the certificate**

iv

**ACKNOWLEDGEMENT**

Apart from my effort, the success of this internship largely depends on the encouragement and guidance of many others. I take this opportunity to express my gratitude to the people who have helped me in the successful competition of this internship.

I would like to thank **Dr**. **N. Siva Prasad,** Pro Vice Chancellor, GITAM Hyderabad and **Dr. CH. Sanjay,** Principal, GITAM Hyderabad

I would like to thank respected **Dr. S.Phani Kumar,** Head of the Department of Computer Science and Engineering for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present an internship report. It helped me a lot to realize what we study for.

I would like to thank the respected faculties **Mr. M. Venkateswarlu** who helped me to make this internship a successful accomplishment.

I would also like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Pulaparthi Sai Kiran

(221710310051)

**Table of content**

|  |  |  |
| --- | --- | --- |
|  | **Title** | **Page No.** |
|  | List of tables………………………………………..  List of Figures |  |
| **1** | Full stack development(introduction).............................. | 9 |
|  | **1.1** steps to connect .html file to the database |  |
|  | **1.2** Scripting |  |
|  | **1.3** Database |  |
| **2** | System requirements………………………………….. | 10 |
|  | **2.1** hardware requirements |  |
|  | **2.2** software requirements |  |
| **3** | Project name………………………………………. | 11 |
|  | **3.1** online music system |  |
|  | **3.2** technologies used |  |
|  | **3.3** technical details |  |
| **4** | Modules………………………………………………... | 12 |
| **5** | Workflow………………………………………………. | 13-38 |
|  | **5.1** workflow of feedback form in music.html | 13-27 |
|  | **5.2** workflow of registering a new user and logging a new user by checking details from database | 27-38 |
| **6** | maintenance…………………………………………….. | 39 |
| **7** | Future scope and future enhancement………………….. | 39 |
| **8** | conclusions……………………………………………...  Bibliography | 39 |

**List of Figures**

**Figure 4.1**  sample authentication……………………………………..

**Figure 5.1.1** using database……………………………………………..

**Figure 5.1.2**  creating a music\_website table ……………………………...

**Figure 5.1.3**  viewing music\_website table…………………………………..

**Figure 5.1.4** view working………………………………………………...

**Figure 5.1.5**  folders and files in visual studio code………………………..

**Figure 5.1.6**  server.js file code…………………………………………….

**Figure 5.1.7**  db.config.js file code………………………………………….

**Figure 5.1.8**  music.controller.js code file…………………………………..

**Figure 5.1.9**  music.controller.js code file continuation……………………..

**Figure 5.1.10** music.controller.js code file continuation……………………….

**Figure 5.1.11** music.controller.js code file end……………………………….

**Figure 5.1.12**  db.js code file……………………………………………………..

**Figure 5.1.13**  music.model.js code file………………………………………….

**Figure 5.1.14** music.model.js code file continuation…………………………..

**Figure 5.1.15** music.model.js code file continuation……………………………...

**Figure 5.1.16**  music.model.js code file end……………………………………..

**Figure 5.1.17**  music.routes.js code file………………………………………...

**Figure 5.1.18**  checking if node.js files are running properly or not…………...

**Figure 5.1.19**  checking via POST requests in Postman………………………...

**Figure 5.1.20** checking via POST requests in Postman…………………………..

**Figure 5.1.21** checking in visual studio code…………………………………….

**Figure 5.1.22**  checking MYSQL database weather data is inserted or not……...

**Figure 5.1.23** use of AJAX in music.html code………………………………….

**Figure 5.1.24**  sending feedback from webpage………………………………...

**Figure 5.1.25**  checking in visual studio code………………………………..

**Figure 5.1.26** MYSQL database before giving feedback………………...

**Figure 5.1.27**  MYSQL database after giving feedback…………………..

**Figure 5.2.1**  create users table……………………………………………….

**Figure 5.2.2**  node.js files in visual studio code……………………………….

**Figure 5.2.3**  server.js file code file………………………………………….

**Figure 5.2.4**  server.js file code file continuation…………………………….

**Figure 5.2.5**  server.js file code file end……………………………………..

**Figure 5.2.6**  checking working of server.js file………………………………....

**Figure 5.2.7** POST request via Postman……………………………………...

**Figure 5.2.8**  checking in Visual Studio Code………………………………….

**Figure 5.2.9**  checking in MYSQL database………………………………….

**Figure 5.2.10**  API of register.html page using AJAX………………………….

**Figure 5.2.11** register from register.html page………………………………….

**Figure 5.2.12**  checking in Visual Studio Code………………………………….

**Figure 5.2.13**  checking in MYSQL database…………………………………….

**Figure 5.2.14**  server.js file code extension file…………………………………..

**Figure 5.2.15**  POST request for valid username and password………………..

**Figure 5.2.16** response for valid username and password…………………….

**Figure 5.2.17** POST request for invalid username and password………………...

**Figure 5.2.18**  response for invalid username and password……………………...

**Figure 5.2.19**  checking in Visual Studio Code…………………………………...

**Figure 5.2.20** API of login.html page using AJAX……………………………….

**Figure 5.2.21**  entering valid details from login.html page………………………..

**Figure 5.2.22**  redirection to music.html page after valid login………………….

**Figure 5.2.23** checking in Visual Studio Code…………………………………...

**Figure 5.2.24**  login details from the database…………………………………….

**Figure 5.2.25** entering invalid details from login.html page…………………….

1.**Full stack development(Introduction)**

**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 Online Music System : we can listen to both online and offline music**

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,AJAX

**Team Size:3**

|  |  |
| --- | --- |
| **NAME** | **WORK DONE** |
| Pulaparthi Sai kiran(221710310051) | Backend implementation of,**login.html** ,**register.htm**l and **music.html** web pages. |
| K Laxmi Swarni(221710310035) | Front-end implementation on **login.html** and **register.html** web pages. |
| CH Priyanka(221710308014) | Front-end implementation on **music.html**  web pages. |

**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 .

****

**Fig 4.1(sample authentication)**

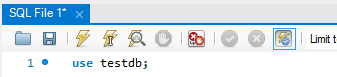
**5. Workflow**

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

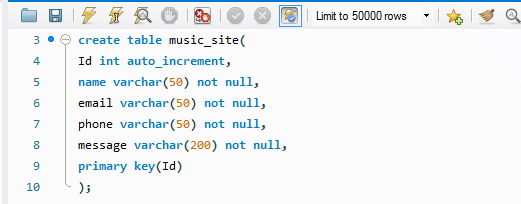
These screenshots will clearly explain how to connect feedback data to the MYSQL workbench.

**Backend of music.html page.**

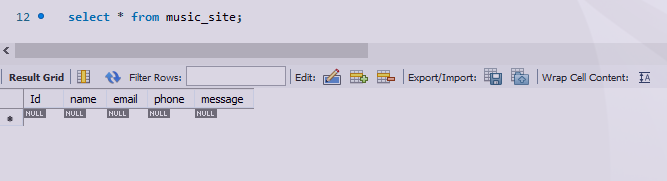
**!!STEP 1!!**



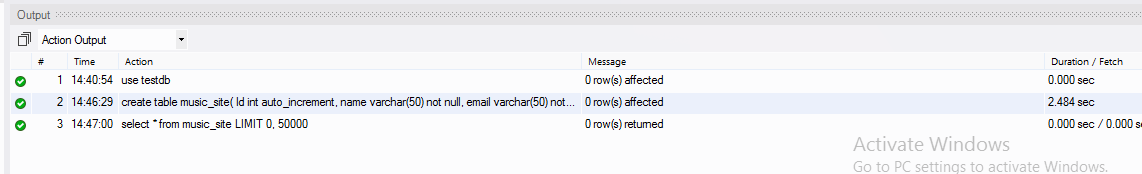
**Fig 5.1.1(using database)**



**Fig 5.1.2 (creating a music\_site table)**



**Fig 5.1.3(viewing the music\_site table)**



**Fig 5.1.4(viewing working)**

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

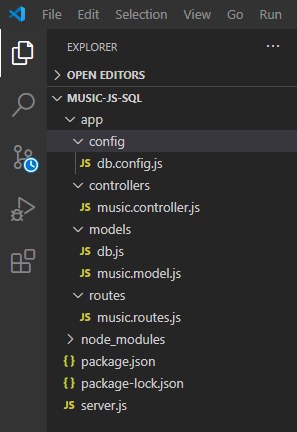
**npm install**  command is used to install node js on visual studio code

**!!STEP 2!!**

Files in node.js:

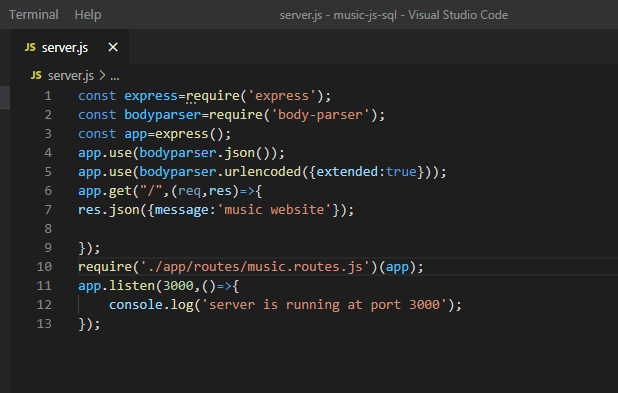
Code of node.js files.

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



**Fig 5.1.5(folders and files in visual studio code)**

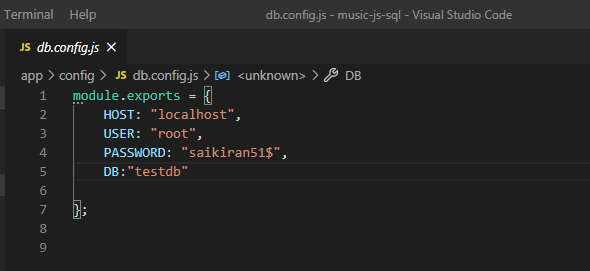
**server.js**

 **Fig 5.1.6(server.js file code)**

In app folder:

config->

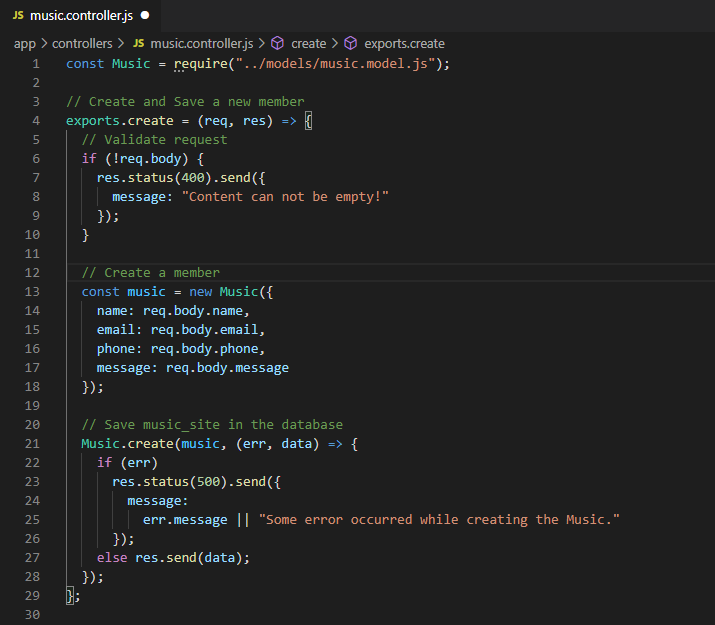
**db.config.js**



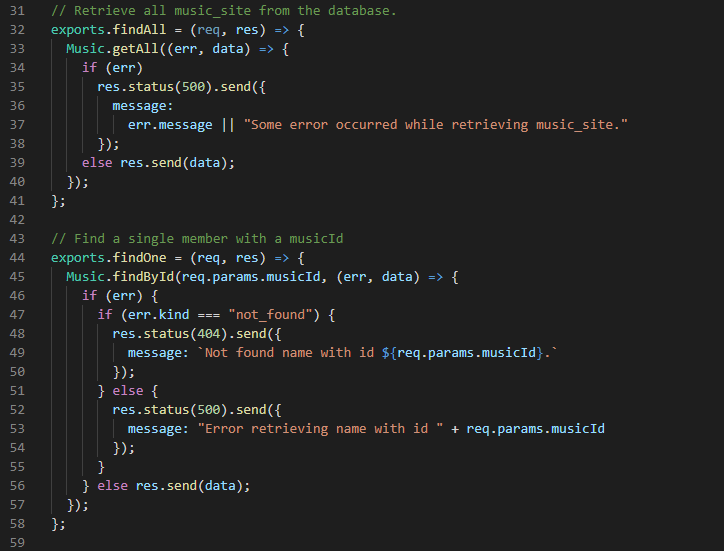
**Fig 5.1.7(db.config.js file code)**

controllers->

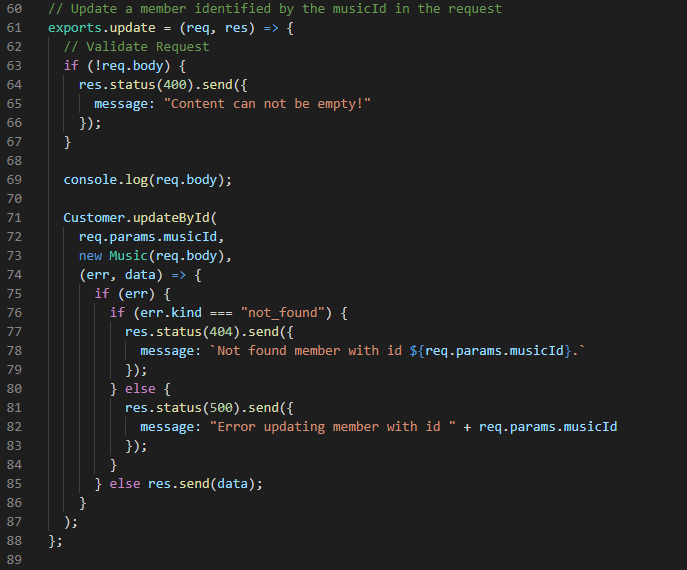
**music.controller.js**



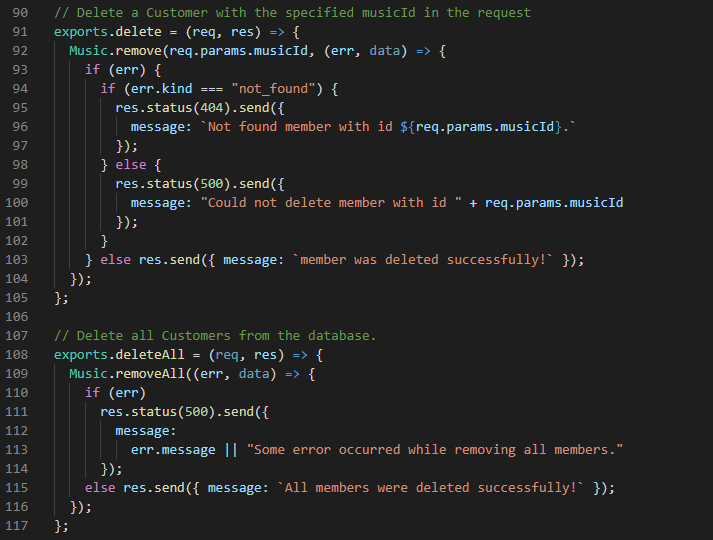
**Fig 5.1.8(music.controller.js file)**



**Fig 5.1.9(music.controller.js file)**



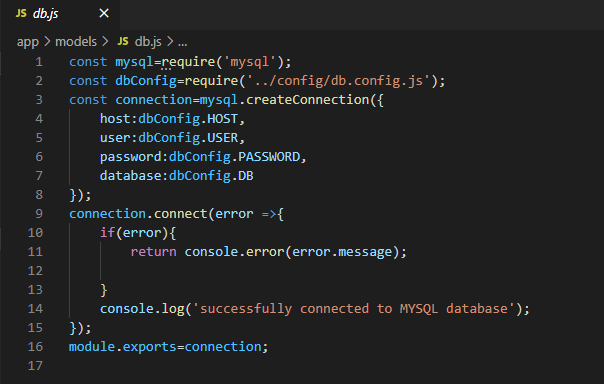
**Fig 5.1.10(music.controller.js file)**



**Fig 5.1.11(music.controller.js file)**

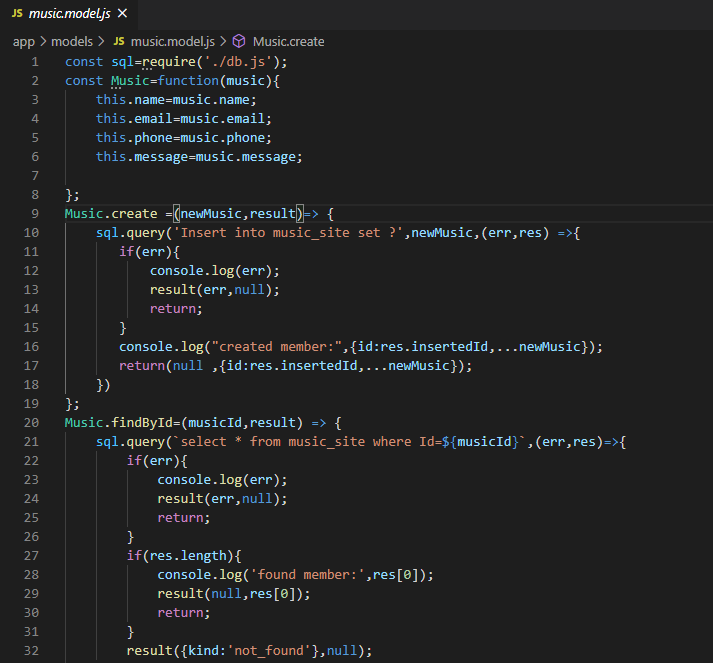
models->

**db.js**

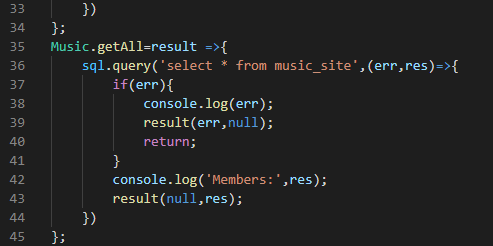


**Fig 5.1.12(db.js file)**

**music.model.js**

****

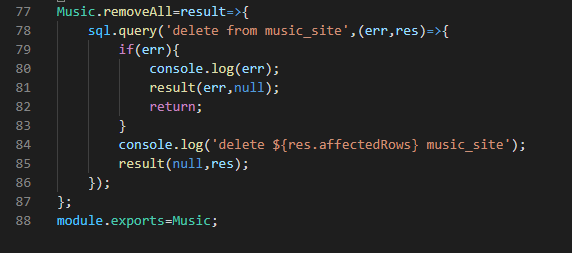
**Fig 5.1.13(music.model.js file)**

****

**Fig 5.1.14(music.model.js file)**

****

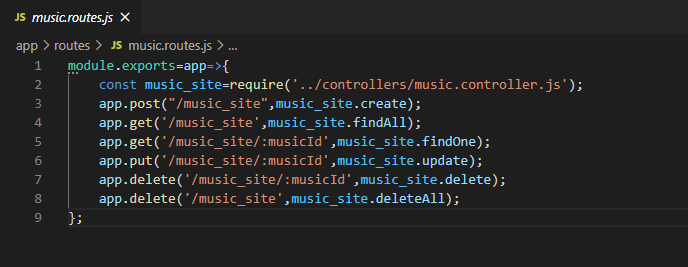
**Fig 5.1.15(music.model.js file)**



**Fig 5.1.16(music.model.js file)**

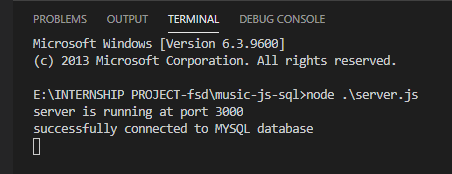
**routes->**

**music.routes.js**

****

**Fig 5.1.17(music.routes.js file)**

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

****

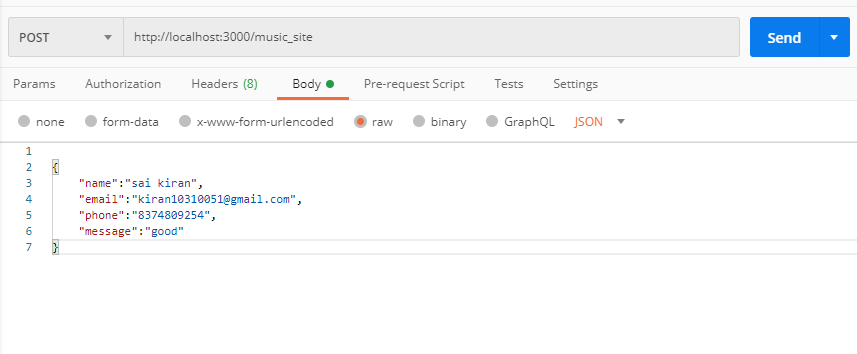
**Fig 5.1.18(checking whether code is running properly or not)**

**!!STEP 3!!**

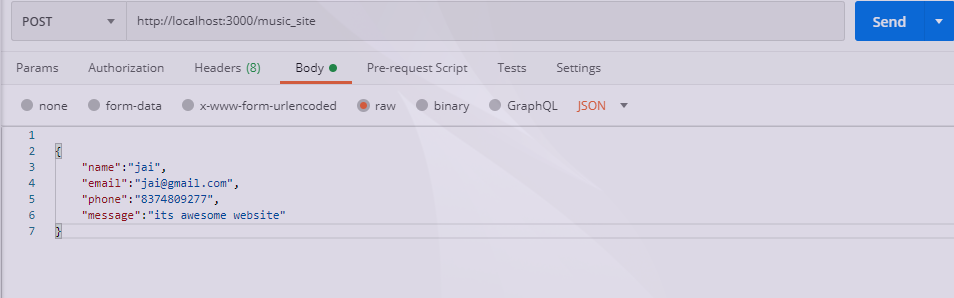
We use postman for testing purposes, whether POST requests are running properly or not.

**In postman:**

after creating post request:

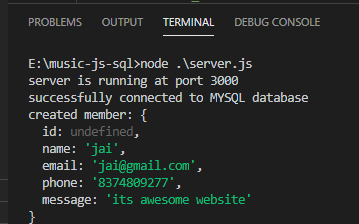


**Fig 5.1.19(checking via POST requests in Postman)**



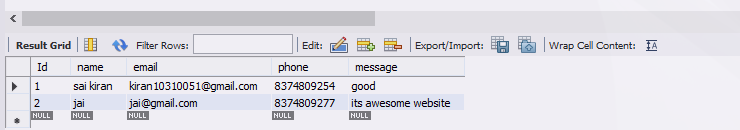
**Fig 5.1.20(checking via POST requests in Postman)**

In visual studio code:



**Fig 5.1.21(checking in visual studio code)**

Sql database after updation:



**Fig 5.1.22(checking MYSQL database weather data is inserted or not)**

**!!STEP 4!!**



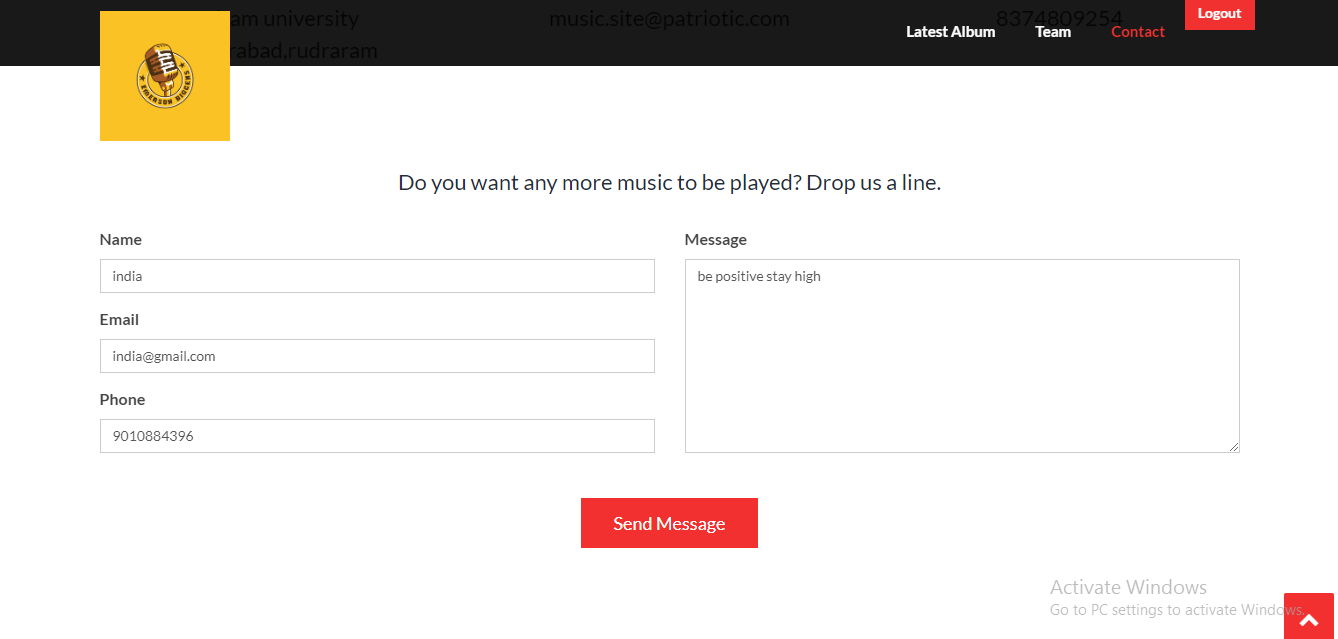
**Fig 5.1.23(use of AJAX in music.html code)**

**!!STEP 5!!**

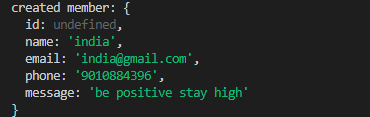
Now inserting values to “testdb” database, to SQL

Open **music.htm**l file

Now sending message to the server side:



**Fig 5.1.24(sending feedback from webpage)**



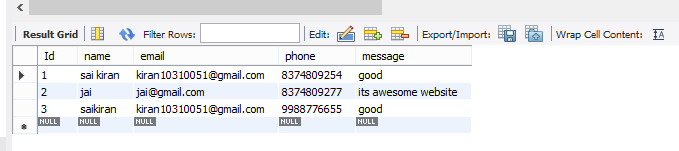
**Fig 5.1.25(checking in visual studio code)**

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:

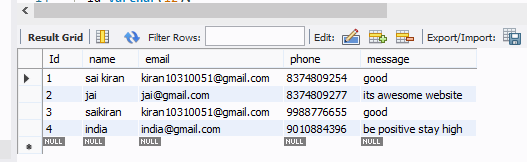


**Fig 5.1.26(MYSQL database before giving feedback)**

Now checking values are inserted or not:



now , the updated database is:



**Fig 5.1.27(MYSQL database after giving feedback)**

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

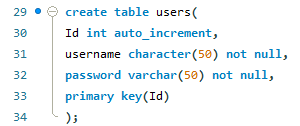
**5.2 workflow of registering a new user and logging a new user by checking details from the database.**

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

**Backend of registration page:**

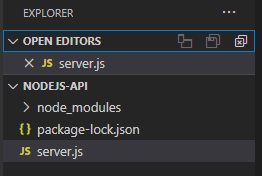
Create table in MYSQL:





**Fig 5.2.1(create users table)**

nodejs-API file on registration page:

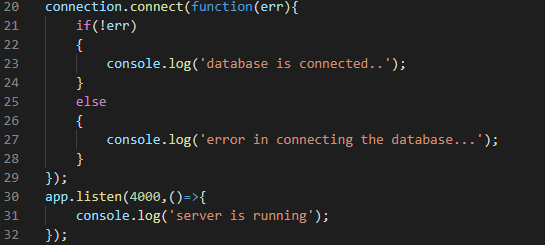


**Fig 5.2.2(node.js files in visual studio code)**

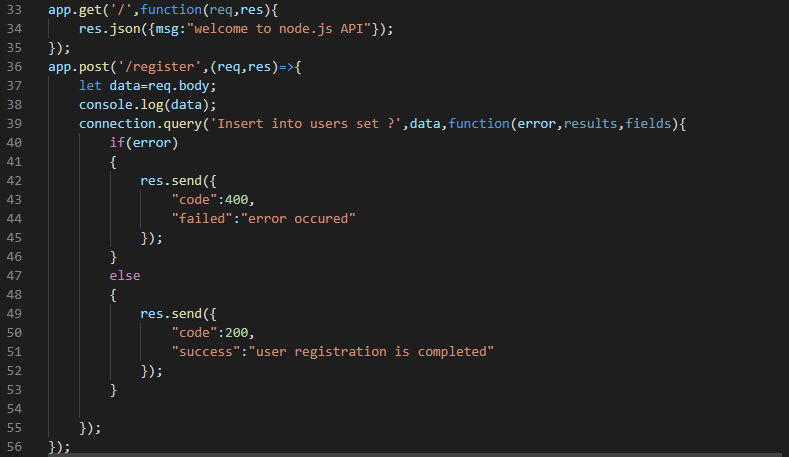
In **server.js** file:



**Fig 5.2.3(server.js file)**

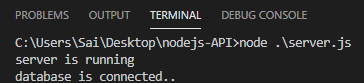


**Fig 5.2.4(server.js file)**



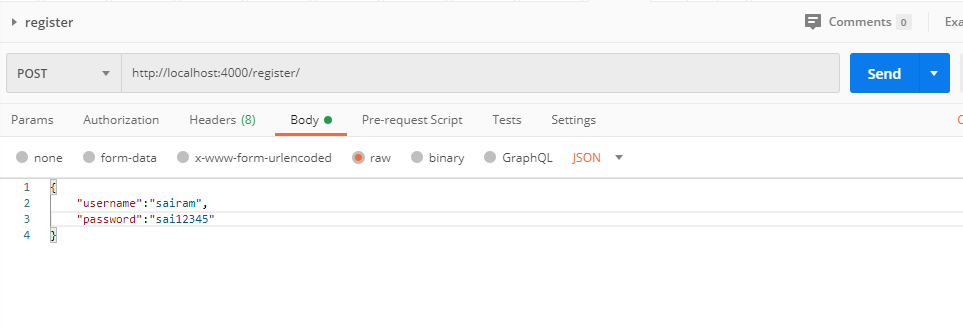
**Fig 5.2.5(server.js file)**

Now check whether code is running properly or not:



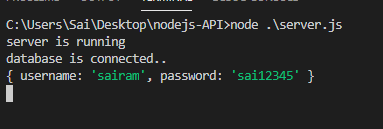
**Fig 5.2.6(checking working of server.js file)**

Now checking with postman:

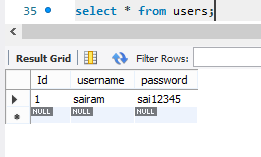


**Fig 5.2.7(POST request via Postman)**

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



**Fig 5.2.8(checking in visual studio code)**



**Fig 5.2.9(checking in MYSQL database)**

Data is inserted successfully:

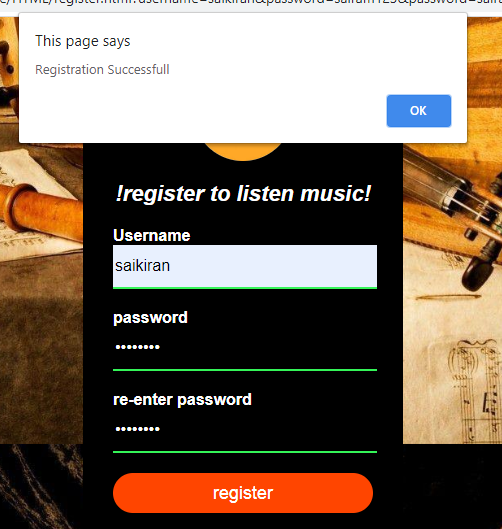
Now inserting data from **register.html** page:

Give **API** connection to **register.html** page:



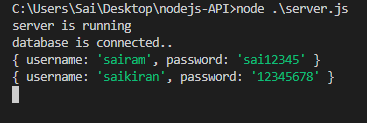
**Fig 5.2.10( API of register.html page using AJAX)**

Now, inserting the data from **login.html** page:

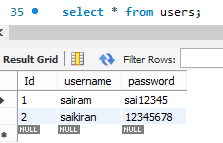


**Fig 5.2.11(register from register.html page)**

Now checking the database part:



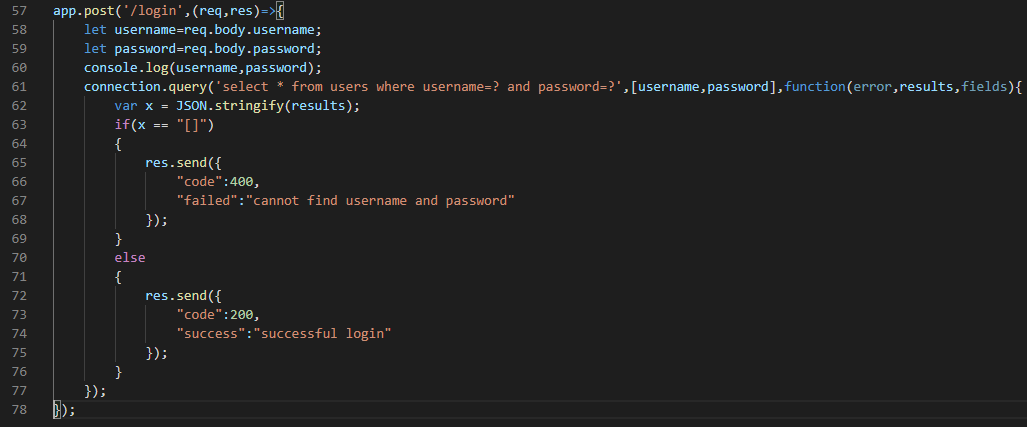
**Fig 5.2.12(checking in Visual Studio Code)**

**Fig 5.2.13(checking in MYSQL database)**

Data is successfully inserted into the database

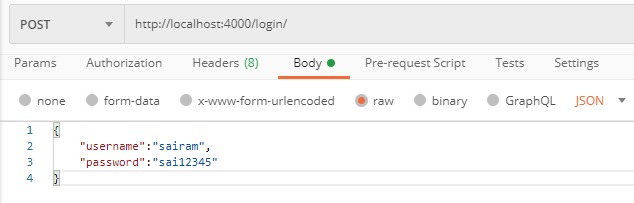
**Back-end of login.html page**

In **server.js** file:

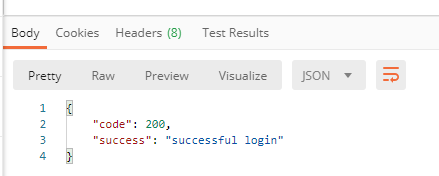


**Fig 5.2.14(server.js file code extension)**

Now checking in postman whether data is validated or not:

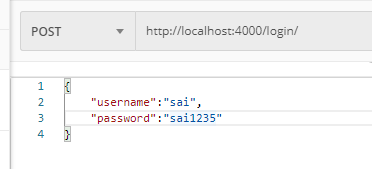


**Fig 5.2.15(POST request for valid username and password)**



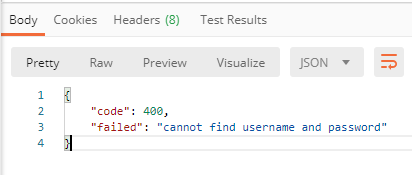
**Fig 5.2.16(response for valid username and password)**

Now inserting some invalid data



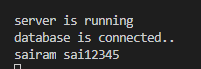
**Fig 5.2.17(POST request for invalid username and password)**

We can see the the entered data is invalid



**Fig 5.2.18(response for invalid username and password)**

Now checking the database part:



**Fig 5.2.19(checking in Visual Studio Code)**

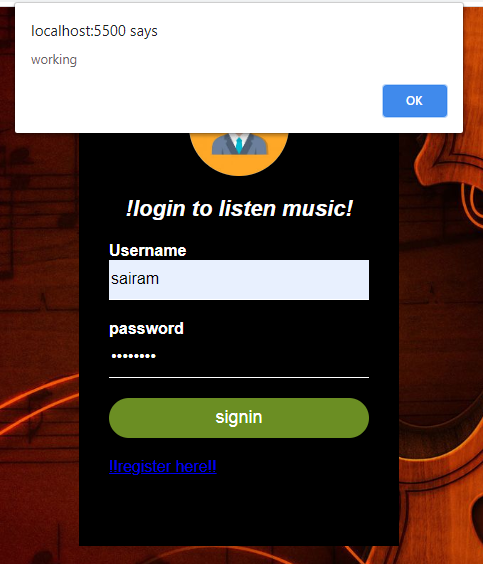
Now connecting API with login.html page:



**Fig 5.2.20( API of login.html page using AJAX)**

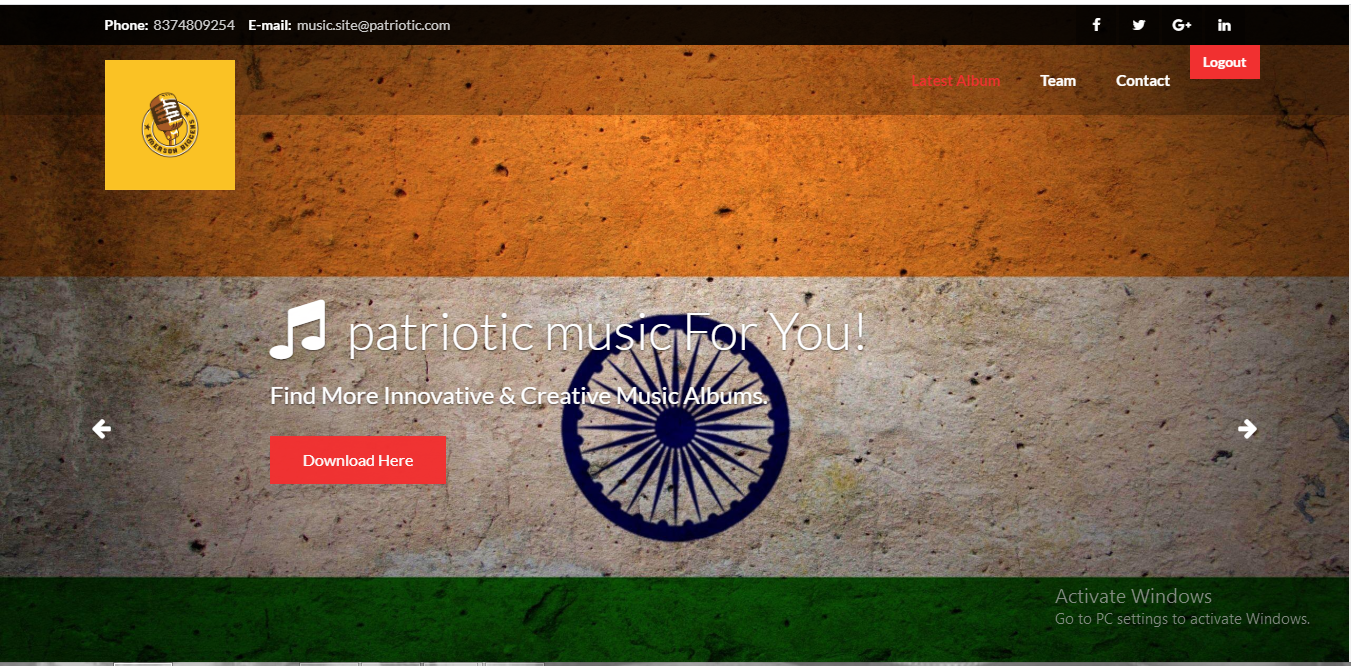
Now entering the dat in login.html page:

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

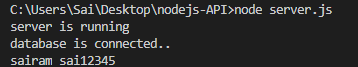


**Fig 5.2.21(entering valid details from login.html page)**

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



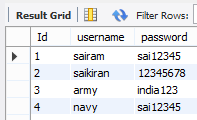
**Fig 5.2.22(redirection to music.html page after valid login)**



**Fig 5.2.23(checking in Visual Studio Code)**

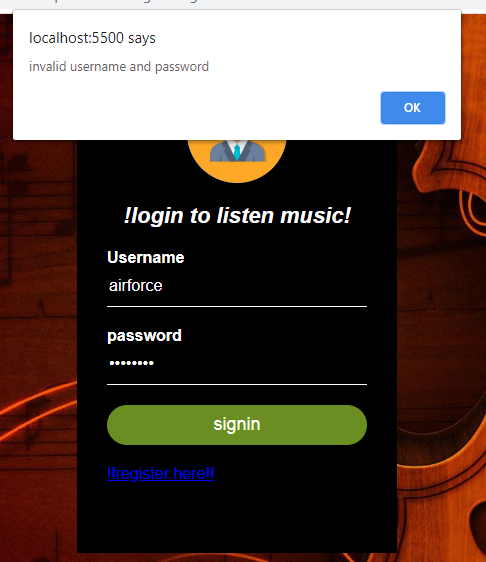
Now giving invalid details which is not present in the MYSQL database

Table shows the data which is already present.



**Fig 5.2.24(login details from the database)**

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



**Fig 5.2.25(entering invalid details from login.html page)**

**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.

**BIBLIOGRAPHY**

1. [www.javatutpoint.com](http://www.javatutpoint.com)

2. [www.w3schools.com](http://www.w3schools.com)

3. [www.getbootstrap.com](http://www.getbootstrap.com)

4. [www.codeigniter.com](http://www.codeigniter.com)

5. [www.stackoverflow.com](http://www.stackoverflow.com)

6. [www.fontawesome.io](http://www.fontawesome.io)

7. <https://nodejs.org/en/>

8.[www.mysql.com](http://www.mysql.com)

9.wikipedia