1. Our back-end part is mysql database schema and all data is in one submitted sql file. Our front-end part is using php and wamp to run in localhost. Firstly, because wamp only works for windows computer, if your computer is max, please use the xampp. The following instruction is in windows. Firstly, you should download the wamp: <http://edu.metinfo.cn/upload/file/wamp-64b.zip> and if your computer has mysql, please change the port of wamp to another port not 3306, please follow directions: <https://stackoverflow.com/questions/37262216/mysql-server-5-7-and-wampserver-in-the-same-machine-wamp-cant-run> And you can extract our zip “Guitar\_Website\_Front\_End” into your www file that is in your wamp file. Later, you should open our “Guitar\_Project.sql” file in your MySQL and run it to create database and you should open “Connect\_Database.php” to change the password and user-name of your database connection. Next, you can open browser on this url: <http://localhost/Guitar_Website/> , click [DIR] Documents, and click Start\_Guitar\_Database to start the guitart website.

**This part is all operations user can do in our front end website:**

After clicking file “Start\_Guitar\_Database” in document dir, you will see home page.

1. Sign Up:
2. click on sign up button
3. choose one type of four user type to sign up ( if you do not want to sign up, click back button)
4. For normal: input any string for name, password, country, state, street, two int for phone and zip code, click submit or jump to choose type page
5. For company: input any string for name, password, country, state, street, two int for phone and zip code, try this company name: “Martin” and security code “AdqwF”, click submit, if succeed, see mention information, if fail, see failed information.
6. For store: input any string for name, password, country, state, street, two int for phone and zip code, try this store name: “Guitar Center” and security code “EDADASF”, click submit, if succeed, see mention information, if fail, see failed information.
7. For museum: input any string for name, password, country, state, street, two int for phone and zip code, try this museum name: “museunA” and security code “132143”, click submit, if succeed, see mention information, if fail, see failed information.

For above sign up page, when submitting one of four type user: it will call procedures for

‘signUpNormal’ (using trigger ‘insertANormal’), signUpCompanyR, signUpStoreR and signUpMuseumR

2. Log In:

1. Click log in buttons: try the following data input
2. Enter name: “second”, password: “123”， and user type : “normal
3. Click login button (this will call the function: logCheck)
4. If you input wrong information, it will show Error: Wrong information, succeed → jump to all operation page

3. After logging in successfully, you will jump to Choose Operations page (Attention: our search company, store and museum information did not work because we spend much time on back end database to create much procedures and tables covering all knowledge we learnt)

1. Search guitar information:
2. click the button: click ‘Look All Guitar’ buttons, you can see all guitars existing in our database (this used showAllGuitars procedure).
3. If you enter brand of guitar and click “search by Brand button”, it will show all guitars that belong to that brand if this brand guitars exist in our database. (this used searchByBrand procedure)
4. If you enter type of guitar following the above type of guitars showing in the page and click “search by Type button”, it will show all guitars that belong to that brand if this brand guitars exist in our database. (this used searchGuitarType procedure)
5. You can click button “back to all operations to continue other operations”
6. In page of “Choose Operations”, you can click button “Show User Information” and will show your user information information according your user type. (this used showOwnUserInfor procedure)
7. In page of “Choose Operations”, you can click button “Edit User Information” , enter new name, password, phone, country, state, street, zip code. And click submit, If edit successfully, it will show success information. Please fill every value. (this used editCommonUserInformation procedure)

**This part is all functionalities we design and create in mysql file but not in website due to limited time and unlearnt knowledge of front end website design: All tests we comment them in mysql file**

4. Show Company Information: given one company name existing in database user want to search , call this procedure to see company information (test: call procedure showCompanyInforByrepresentative(‘Martin’))

5. Show Retail Store Information : given one company name existing in database user want to search , call this procedure to see company information (test: call procedure showRetailInforByrepresentative(‘Guitar Center’))

6. Show Museum Information: given one company name existing in database user want to search , call this procedure to see company information (test: call procedure showMuseumInforByrepresentative(‘museunA’))

7. Edit Company Information: provide necessary security of company and any information you want to edit and others just remain null (test: call procedure editCompanyInfor('AdqwF', 'companyWang', 'anyone', 'https://www.fender.com/', 'ewqfqewa', 312412);)

8. Edit Store Information: provide necessary security of store and any information you want to edit and others just remain null (test: call procedure editStoreInfor('EDADASF', 'storeLi', 'Guitar Center', 'Africa', 'test', 'test', 132324, 31124312, 'test');)

9. Edit Museum Information: provide necessary security of museum and any information you want to edit and others just remain null (test: call procedure editMuseumInfor('132143', 'museumLiu', 'Sphone Hall', 'test', 'test', 'test', 123, 3321, 'test');)

10. Leave comments: After users log in successfully, they can leave comment for guitars. They could only leave one comment and score for each guitar. (test: call procedure leaveComments('first', 'Martin 00-42SC', 'that was one wonderful guitar', 45, 0);)

Trigger: scoreInsert: updater average score when the one comment is updated for one guitar

-- the range of score should become 0 to 100

Trigger scoreUpdate: update score of guitar in comments, calculate the average score of guitar again if there is one comment for the same guitar by the same user

11. Add favourite list:-- To add a guitar to favourite list of normal users. (test call procedure addFavouriteList('second', 'Martin 00-42SC');)

Trigger: favouritelistCheck: trigger for preventing users from repeating adding same guitars to their favorite lists.

12. Search Comment: To search comments for given guitar's name. (test: call procedure searchComments('Martin 00-42SC');)

13. Post Sale: Post the sale info of guitars. It will need company name, store name, and price of guitars. (test: call procedure postSale('Fender', 'Fender CP-100', 'Guitar Center', 40.0);)

Trigger: updateLowest -- A trigger to update the lowest price of the guitar if there's a new sale posted with lower price.

14. Post Exhibition information: -- This is for museums to post an exhibition of guitars. It will need gutarName, museumName, start date, end date and ticket price. (test: call procedure postExhibition('museunA', 'Yamaha CSF60', '2012-01-19', '2013-01-19', 100);)

15. Expired Exhibition : A procedure to check the end date of exhibition drop the expired exhibition.

After use postExhibition procedure to post one information that end date is passed to current time, call expiredExhibition, it will delete it automatically

16. Update Level: A procedure to check the date and update normal users level according to their singup date.drop procedure if exists updateLevel;

(test: insert into users values(6, 'saleGood', 'password', 'normal', '2012-01-19', 012331326, 'US', 'MA', 'C st.', 01000);

Call procedure updateLevel();)

17. Guitar\_engine: A event call guitar\_name\_given, guitar\_name\_given every day.

Please use set global event\_scheduler = on; We have tested updateLevel and expiredExhibition successfully, so we think this event also works.

18. createAGuitar: A procedure to insert a guitar information. (test: call procedure createAGuitar('Martif', 'Classic', '00-42SC', 'Martin', 'This parlor guitar is born out of the success of the limited edition (25 pieces) 00-45SC, which were quickly sold out after being launched.',

1, 2, 'Polished Gloss', 'Solid Sitka Spruce', 'Cocobolo');

19. createAComp: A procedure to insert a piece of company information. (test: call procedure createAComp('Taylor\_kingSaff', 'https://www.taylorguitars.com/',

'support@taylorguitars.com', 802182, '321423');)

20. createAStore: A procedure to insert a piece of retail store information. (test: call procedure createAStore('Guitar Power', 'United States', 'MA', '1295 Huntington Ave # 304, Boston', 02115, 614126077, 'Wednesday 10AM–9PM Thursday 10AM–9PM Friday 10AM–9PM Saturday 10AM–8PM Sunday 11AM–7PM Monday 10AM–9PM Tuesday 10AM–9PM', 'EDADASF');)

21.createAMu: A procedure to insert a piece of museum information. (test: call procedure createAMu('museunWHad', 'USA', 'MA', 'A street', 021132, 00131, '9:00 - 18:00', '13afds3');)

22.deleteComments: A procedure to delete the certain comment. Users will specify the guitar name of the comment they want to delete. (test: call procedure leaveComments('saleGood', 'Martin 00-42SC', 'that was one wonderful guitar', 45, 0);)

23. deleteUser: A procedure to delete the user account. Account could be all types of users.