DBMS PROJECT

HOSTEL QUERY MANAGEMENT

HOSTEL QUERY MANAGEMENT SYSTEM it is software which is helpful for students who are residing in hostels.

Over the years, Educational Institutions are rapidly increasing thereby hostels are also increasing for the accommodation of students studying in these Universities/colleges etc...

Lots of students are facing some problems regarding room, lift etc.. Here comes our website

This helps to deal with the problems faced by students/staff/anyone who is staying in hostels and avoids the problems which occur when carried manually

We here developed a software which helps people(students/staff/non-staff) who are having issues or queries can solve their issue by just entering in the website.

SYNOPSIS

This project has a large scope as it has the following features which help in making it easy to use, understand and modify it: -

- ➔ Easy to online drafting.
- ➔ No Need to do Paper Work.
- ➔ To save the environment by using paper free work.
- ➔ To increase the accuracy and efficiency of the placement procedure.
- ➔ Management of Student Data.

Tools & Technology:

- ➔ **NodeJS**: - Nodejs is a programming language used in all kinds of software but majorly used for back-end scripting, It's originated from chrome's JavaScript engine known as "V8" later-on it was implemented in all the operating systems for programming/application means
- ➔ We have used this NodeJs to connect the MySQL to User end-face, NodeJS plays a major role in connecting the user to the backend stage and access all the previous query that has posted.
- ➔ **MySQL**: - MySQL is a RDMS used to manage and operate the data from user to backend. This is where all data is collected and displayed in dynamic webpages.
- ➔ **ExpressJS**: - It is a minimal and flexible Node.js web application framework that provides a set of features for web and mobile applications.

Project flow

Files path: -

HostelQueryManagement/

|__controllers/

| |__auth.js

|__public

|__**routes/**

| |__auth.js

| |__pages.js

|__**views/**

| |__admin.ejs

| |__data.ejs

| |__index.ejs

| |__Queryregister.ejs

| |__s_status.ejs

| |__slogin.ejs

| |__status.ejs

|__**app.js**

|__**database.js**

|__**package.json**

|__**package-lock.json**

Package and modules required: -

- ➔ NodeJS
- ➔ MySQL
- ➔ EJS
- ➔ Npm install mysql,ejs,css,html,i //type in CMD

APP.JS

App.js contains all the modules required for starting the server we imported MySQL library/module, view engine is set to express JavaScript.

- ➔ At initial when the server is started on the localhost the app.js connects to the MySQL and logs “Database connected!”
- ➔ Later when the user tries to access the /auth the router folder is imported.

Routes/pages.js

In pages.js all the links are redirected here when the user access /auth/” <PAGE>”. In link completion is done in auth.js

- ➔ We used pages.js in a separate folder since there are many pages to forward to the user and sensitive information is passed to the database, we used to redirect to the controllers for all MySQL operations.

Routes/auth.js

In auth.js we wait to revive the data from controllers were all the MySQL operations are performed for better security and reduce the code complexity

Controller/auth.js

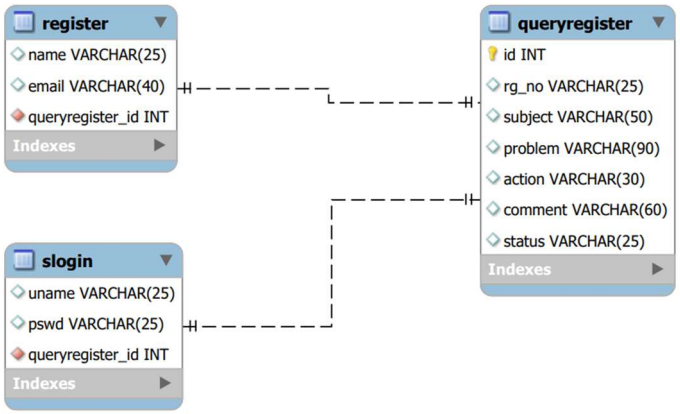
Here the connection is made with MySQL server on port 3306 using the MySQL instance and receiving the data for the user input for the express java script webpages. In total there are 5 pages which needs information to pass though these script in order complete there MySQL operations

- ➔ After the quires are registered the respective webpages is to be redirected for further process.

Views/

- ➔ Index.ejs is the home page which has 2 buttons for admin login and student login
- ➔ Slogin.ejs is the login page for students
- ➔ Querregister.ejs is the page where all students register their quires which is stored in database.
- ➔ S_status.ejs is the webpage where student can see his/her posted query status and updates.
- ➔ Admin.ejs is the login page for warden/hostel manager for resolution for query’s solutions
- ➔ Data.ejs id the table page where all the problems are notified and action is taken in a_status.ejs

ER-DIAGRAM



Register

CREATE TABLE IF NOT EXISTS `dbmsproject`.`register` (`name` VARCHAR(25) NULL DEFAULT NULL,

`email` VARCHAR(40) NULL DEFAULT

Constraints used: -

- ➔ Name varchar(25)
- ➔ Email varchar(40) as password

Table register contains User name and the password for the warder/Admin to Login and check the Problems registered students and update the status of each query posted.

Slogin

CREATE TABLE IF NOT EXISTS `dbmsproject`.`slogin` (

`uname` VARCHAR(25) NULL DEFAULT NULL,

`pswd` VARCHAR(25) NULL DEFAULT NULL);

Constraints used: -

- ➔ Uname varchar(25)
- ➔ Passwd varchar(40) as password

Slogin table is used for the student login for registering the query/ problem the student has and he also can view the st using the query register page.

QUERY REGISTER

CREATE TABLE IF NOT EXISTS `dbmsproject`.`queryregister` (

`id` INT NOT NULL AUTO_INCREMENT,

`rg_no` VARCHAR(25) NULL DEFAULT NULL,

`subject` VARCHAR(50) NULL DEFAULT NULL,

`problem` VARCHAR(90) NULL DEFAULT NULL,

`action` VARCHAR(30) NULL DEFAULT NULL, `comment` VARCHAR(60) NULL DEFAULT NULL, `status` VARCHAR(25) LL DEFAULT NULL, PRIMARY KEY (`id`));

CONCLUSION

As the project we choose is HOSTEL QUERY MANAGEMENT SYSTEM.

We came up with a different approach than others by using Node-JS for connecting MySQL and browser for displaying data.

As we used NODEJS we declared the localhost server in our system and got the fina output as we expected.

In this, we connected

MYSQL-Route --> NODEJS-Compiler--> WEB-Browser.

ALGORITHM